

extended. If the predefined policy 240 mandates the extension of the file system, the manager 20 identifies which LUNs should be utilized and assigns one or more identified LUNs to that host 12. Upon receiving the LUN IDs from the manager 20, the agent 24 that is operating on the host 12 extends the file system as discussed below.

5

Generally, the agent 24 executes the following steps to extend the file system:

1. Initialize the newly assigned LUNs by converting them to a form understood by the host operating system. In some operating systems, this is referred to as writing a signature to the devices and is analogous to formatting a hard disk.
2. Create a logical representation (e.g., an "object") of each newly assigned LUN that corresponds to the underlying physical devices.
3. Add the objects to the logical grouping that form the host file system.
4. Increase the logical volume size of the file system by an amount equal to the entire size of the newly added LUNs.
5. Increased the size of the file system to occupy the increased logical volume size

In an embodiment of the invention for use with the AIX Journal file system, the agent extends the host file system by executing the steps of

1. Convert the newly assigned LUNs to physical volumes using built-in host operating system features.

5      2. Add the physical volume(s) into the volume group of the file system to be extended, using the host API.

3. Extend the logical volume onto the new assigned LUNs using the host API.

10      4. Extend the file system by an amount equal to the capacity of the newly assigned LUNs, again, using the host API.

15      Upon completion, the agent 24 notifies the manager 20 of the successful file extension (and, subsequently, the user). If any of the above steps fail, the file system is not extended and the agent 24 notifies the manager 20 of the failure.

20      In an embodiment of the invention for use with, Veritas or Unix file systems (created using Veritas volume manager and managed under a Solaris operating system) are dynamically extended. As above, once given the newly assigned disk ID(s) and the name of the file system to extend, the agent 24 automatically increases the file system and its underlying volume by an amount equal to the size of the assigned disks. The specific steps (analogous to those above) that the agent 24 performs to accomplish this task are as follows.

1. The agent utilizes the host Solaris API to initialize the LUNs by writing a new label to the newly assigned LUNs (this equates with writing a signature to the disks).
2. The agent utilizes the Veritas API to configure the LUN(s) for use with Veritas Volume manager by converting the newly assigned LUN(s) into VM Disks (which are analogous to physical volumes).
3. The agent utilizes the Veritas API to add the VM Disk(s) to the disk group where the logical volume of the file system to be extended resides.
4. The agent utilizes the Veritas API to increase the size of the file system and its underlying volume by adding all the available disk space from the assigned LUN(s).

As above, if any of the steps fail, the file system is not extended. The manager 20 is notified of the success or failure of the file extension procedure.

FIGURE 35 illustrates the process 308 that the agent 24 undertakes to extend file systems in accord with the invention. First, a new label is written to an assigned LUN 310. The newly labeled LUN 312 is then initialized and configured for use with the Veritas volume manager (VM) by converting the LUN 312 into VM disks 314. This involves separating the LUN 314 into one or more partitions (in this example having a total size of 2 Gigabytes). The configured disk 322 is then added to a disk group 316 that contains the file system to be extended. In this example, disk group 316 already contains two volumes 318, 320, and the file system to be